

Avoiding the Flip Ambiguities in 2D Wireless Sensor Localization by Using Unit Disk Graph Property

Onur Çağırıcı

Technical Report

Abstract

In this paper, we propose a range-based localization algorithm to localize a wireless sensor network (WSN) in 2D. A widely used algorithm to localize a WSN in 2D is trilateration, which runs in polynomial time. Trilateration uses three distance measurements to localize a node. In some cases, the lack of connectivity leads to a low percentage of localized nodes since a node's position can be fixed using three distance measurements. We propose an algorithm that finds the position of a node by using the absence of a distance measurement in addition to a third distance measurement. If two nodes are not able to sense each other, that means the distance between them is more than the sensing range. Therefore, our algorithm checks if the possible positions of an unlocalized node u is inside the sensing range of a localized node ℓ that is not the neighbor of u . In such case, we eliminate one of the possible positions.

1 Introduction

One of the main challenges in WSN applications is the *localization problem* [1, 2, 3, 4, 5, 6]. *Localization* or *positioning* of a WSN is determining the positions of the sensor nodes with respect to each other. Even though GPS is a very powerful tool to determine the position of an object, it is usually not efficient enough when the sensor nodes are too close to each other. Considering the cost and the efficiency of the system, *range-based localization* is widely used to determine the positions of the sensor nodes. Range-based localization is determining the relative positions of the node by only using the measured pairwise distances [7]. The localization process is carried out by using a weighted and undirected graph $G = \langle V, E, W \rangle$, called a *WSN graph*, where each sensor node correspond to a vertex $v \in V$, the pairwise connection between two nodes v

and w correspond to an edge $\{v, w\} \in E$ and the computed distance between these nodes correspond to the weight $w(e); e \in E$ of the edge in the graph. A WSN graph satisfies the property of unit disk graph (UDG). In a UDG, an edge $\{v, w\} \in E$ exists if and only if the Euclidean distance between v and w is less than or equal to a specific value called *sensing range*. Saxe [8] showed that localization of a graph in \mathbb{R}^d is NP-complete where $d \in \mathbb{Z}^+$, which is closely related to the embeddability problem. Aspnes *et. al.* than proved that the problem is NP-complete in *unit disk graphs* as well [9].

In order to find the position of a point in \mathbb{R}^2 , we need to know the distance of that point to three non-collinear points. Within the scope of WSN localization, the points are the sensor nodes (or the vertices of the WSN graph), and the distances are the measured distances between sensor nodes (or the weighted edges of the WSN graph). If there exists an ordering in a WSN graph such that beginning from three points, we are able to determine the positions of the rest, then we can localize all the nodes in the graph in polynomial time, using an algorithm called *trilateration* [10]. The ordering, hence, is called *trilaterative ordering* and WSN graph that contains such ordering is called *trilateration graph* [11].

We exploit UDG property to eliminate one of the possible positions for an unlocalized node that has two localized neighbors. Instead of looking for a third distance measurement to localize a node, we forbid a node to be in the area that is covered by the sensing ranges of the localized nodes that are not neighbors of the unlocalized node. We show such a case in Figure 1.

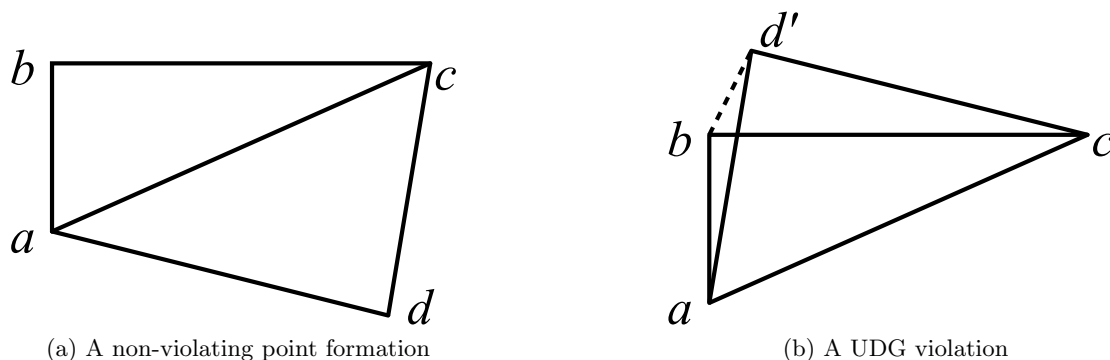


Figure 1: When we fix the positions of a , b and c , (a) does not lead to a violation but (b) is a violation

In Figure 1(a), we see four nodes with five pairwise distances. By definition, the graph is not globally rigid and not localizable. However, by a simple reasoning, we can observe

that the point formation shown in Figure 1(b) is violating the UDG property because there is no such edge $\{b, d'\}$.

2 Assumptions and Terminology

We have mentioned the concepts of WSN graph and UDG in the previous section. Before moving onto the algorithm, let us introduce the assumptions and terminology that we use throughout the manuscript in this section.

We assume that all the sensors are fully functional and identical i.e. the graphs that we deal with are UDGs. This assumption forms the base of our algorithm. The neighbor list $N(i)$ of a node i contains another node j if and only if the Euclidean distance between i and j is less than or equal to the sensing range of the sensors Θ . Neighbor list of a node is an ordered list and j^{th} neighbor of the node i is denoted by $N_j(i)$. The last neighbor in the neighbor list of i is denoted by $N_{last}(i)$. Similarly, a node i has a list of its localized neighbors denoted by $LN(i)$. Following the same notation, $LN_j(i)$ is the j^{th} localized neighbor of i and $LN_{last}(i)$ denotes the most recently localized neighbor of i . The number of localized neighbors of i is also kept and denoted by $i.LNCount$.

The edge weights of the network graph G is determined by adding a random value $\varepsilon \in \mathbb{R}$ to each actual pairwise distance δ_{ij}^{actual} . Thus, the pairwise distances are computed as $\delta_{ij} = \delta_{ij}^{actual} + \varepsilon$, where ε is the randomly gathered noise value unless stated otherwise. The error model will be detailed in Section 5. In order to denote the measured distance between a node i and its j^{th} neighbor, instead of $\delta_{iN_j(i)}$, we use the notation $\delta_{N_j(i)}$.

A *unit disk graph violation* is placing a node i at some position (x, y) such that the Euclidean distance between the position of a localized node $\ell \notin N(i)$ and (x, y) is smaller than or equal to the sensing range.

3 Localization a WSN graph in 2D Using UDG Violations

When a node is being localized, two distance measurements leave us with two possible positions for that node. Hence, we need a third distance measurement in order to fix the position of the node. Considering the noise, we give the function to choose between two possible points in Figure 2.

$$\sigma(p_1, p_2, d) = \begin{cases} \text{NULL}, & \text{if } (|\delta_{jp_1} - d| \leq d * P) \wedge (|\delta_{jp_2} - d| \leq d * P) \\ p_1, & \text{if } |\delta_{jp_1} - d| \leq d * P \\ p_2, & \text{if } |\delta_{jp_2} - d| \leq d * P \\ \text{NULL}, & \text{otherwise} \end{cases}$$

Figure 2: Picking a position among two candidate points

```

LOCALIZEGRAPH( $G$ )
input: WSN graph  $G = \langle V, E, W \rangle$ 
output: 2D point formation of  $G$ 
1:  $\mathbb{F}_{\text{best}} \leftarrow ([ ], [ ])$  /*Best point formation*/
2: for each non-collinear and fully-connected  $\{a, b, c\} \subseteq V$  do
3:    $\mathbb{F}_{\text{current}} \leftarrow ([ ], [ ])$ 
4:   Pick  $a, b$  and  $c$  as seed nodes
5:    $Q_{\text{process}} \leftarrow [a, b, c]$ 
6:   while  $Q_{\text{process}}$  is not empty do
7:      $i \leftarrow \text{dequeue}(Q_{\text{process}})$ 
8:     add  $(i, i.Pos)$  into  $\mathbb{F}_{\text{current}}$ 
9:     for all  $j \in N(i)$  do
10:       $result \leftarrow \text{FALSE}$ 
11:      if  $j.LNCount \geq 2$  then
12:         $result \leftarrow \text{BILATERATE}(j)$ 
13:      if  $j.LNCount \geq 3$  then
14:         $result \leftarrow \text{TRILATERATE}(j)$ 
15:      if  $result = \text{TRUE}$  then
16:         $\text{enqueue}(j, Q_{\text{localized}})$ 
17:   end while
18:   if  $|\mathbb{F}_{\text{current}}^1| = |V|$  then return  $\mathbb{F}_{\text{current}}$ 
19:   if  $|\mathbb{F}_{\text{current}}^1| > |\mathbb{F}_{\text{best}}^1|$  then  $\mathbb{F}_{\text{best}} \leftarrow \mathbb{F}_{\text{current}}$ 
20: end for
21: return  $\mathbb{F}_{\text{best}}$ 

```

Figure 3: Overall localization algorithm

```

BILATERATE( $j$ )
1: Find possible positions  $p_1, p_2$  of  $j$  with respect to  $LN(j)_1, LN_{last}(j)$ 
2:  $violate_1 \leftarrow \text{FALSE}$ 
3:  $violate_2 \leftarrow \text{FALSE}$ 
4: for each localized node  $i \notin N(j)$  do
5:   if  $d(p_1, i) < \Theta$  then  $violate_1 \leftarrow \text{TRUE}$ 
6:   if  $d(p_2, i) < \Theta$  then  $violate_2 \leftarrow \text{TRUE}$ 
7:   if ( $violate_1$ ) AND ( $violate_2$ ) then return FALSE
8: if ( $!violate_1$ ) AND ( $!violate_2$ ) then return FALSE
9: if  $!violate_1$  then  $j.Pos \leftarrow p_1$ 
10: if  $!violate_2$  then  $j.Pos \leftarrow p_2$ 
11: return TRUE

```

Figure 4: Bilateration with missing edges

```

TRILATERATE( $j$ )
1: Find possible positions  $p_1, p_2$  of  $j$  with respect to  $LN_1(j), LN_2(j)$  and  $LN_{last}(j)$ 
2:  $violate_1 \leftarrow \text{TRUE}$ 
3:  $violate_2 \leftarrow \text{TRUE}$ 
4: for each localized node  $i \notin N(j)$  do
5:   if  $d(p_1, i) < \Theta$  then  $violate_1 \leftarrow \text{FALSE}$ 
6:   if  $d(p_2, i) < \Theta$  then  $violate_2 \leftarrow \text{FALSE}$ 
7:   if ( $!violate_1$ ) AND ( $!violate_2$ ) then return FALSE
8: end for
9: if ( $!violate_1$ ) AND ( $!violate_2$ ) then
10:    $j.Pos \leftarrow \sigma(p_1, p_2, LN_{last}(j))$ 
11:   if  $j.Pos$  is NULL then return FALSE
12:   return TRUE
13: end if
14: if  $!violate_1$  then  $j.Pos \leftarrow p_1$ 
15: if  $!violate_2$  then  $j.Pos \leftarrow p_2$ 
16: return TRUE

```

Figure 5: Trilateration with missing edges

4 Localizing a Wheel Graph in Polynomial Time

In this section, we explain how our algorithm can be used to localize a wheel graph in polynomial time. Clearly, we assume that the corresponding graph is a unit disk graph. Aspnes *et al.* proved that localizing a wheel graph is NP-hard [11]. The proof is reduction from SET COVER problem. Given the fact that a WSN graph should satisfy the property of a UDG, we then are able to use our algorithm to localize a wheel graph. If we pick three nodes as seed, then a fourth node can be localized by using two of the seed nodes. Aspnes *et al.* [11] mentions that since we are not able to know which direction to place the nodes of a wheel graph, the problem is hence NP-hard. Considering UDG violations, we are able to fix the positions of the nodes in polynomial time with the algorithm given in Figure 4.

In Figure 6 we see a WSN graph that is localizable but does not have a trilaterative ordering. Since the graph contains many wheel graphs, trilateration cannot localize any of the nodes in this graph. In Figure 7, we give four examples of localization of the graph given in Figure 6 with various values of noise magnitude.

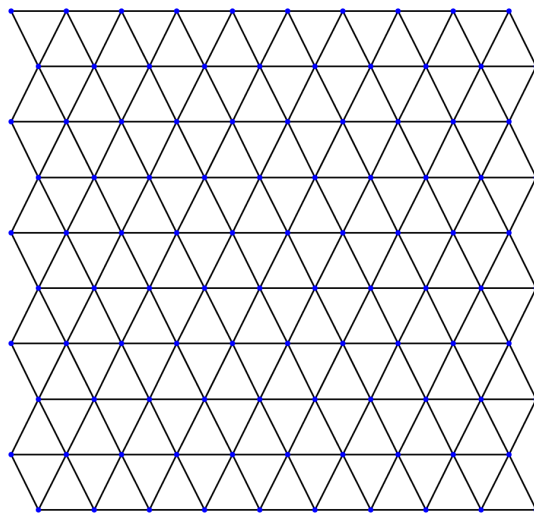


Figure 6: A wheel graph

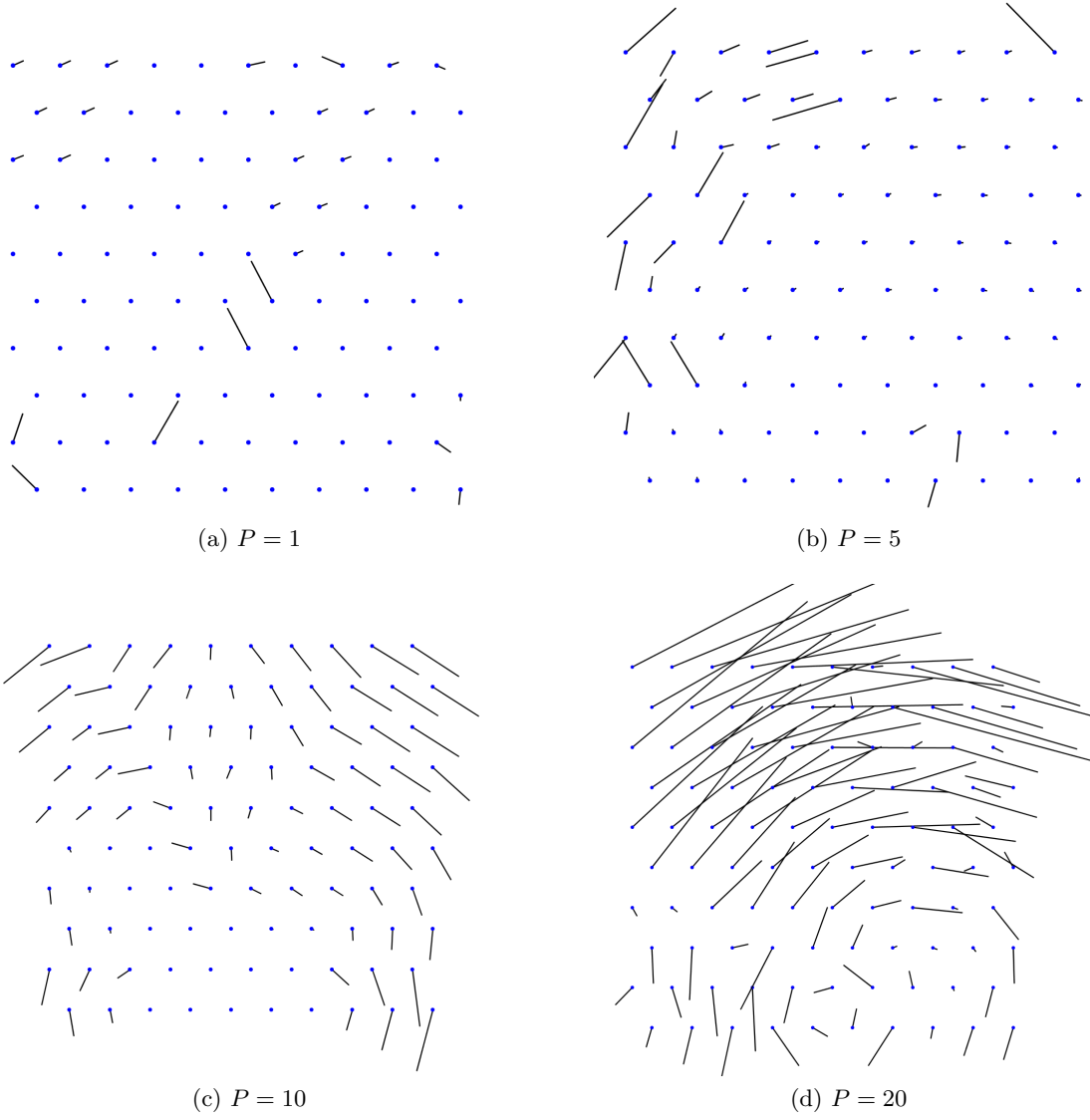


Figure 7: Localization of a WSN that contains many wheel graphs where error magnitude is 1 (a), 5 (b), 10 (c) and 20 (d)

5 Experimental Results

In this section, we conduct experiments to see the how our algorithm improves localization with respect to the mere trilateration. We compare two algorithms first with noiseless distance measurements to see the effect of connectivity on both. Then, we pick the connectivity values where each algorithm reached to a recall percentage greater than 99% and see the effect of environmental noise. The noise is modeled inline with experimentally gathered data in [12, 13]. Each edge $\{v, w\} \in E$ in $G = (V, E)$ is modified with respect to a random number generated using a Gaussian random distribution with $N(f(\Theta), P/100)$ where P is the magnitude of the error and Θ is the sensing range. $f(\Theta)$ is defined as follows;

$$f(\Theta) = 0.022\ln(1 + \Theta) - 0.038$$

In Figure 8 we show the results of tests with noiseless range measurements. The graph tells us that if UDG violations are considered, a WSN graph can be localized with less sensing range. Our algorithm is able to localize more than 99% of the nodes when the connectivity is around 8 whereas mere trilateration requires 15 connectivity per node.

Now, let us show the effect of environmental noise. In Figure 9, we set the connectivity value as 8 units for our algorithm and 15 units for mere trilateration and plot the average offsets of both algorithms. The graph tells us that considering UDG violations causes more offset than mere trilateration.

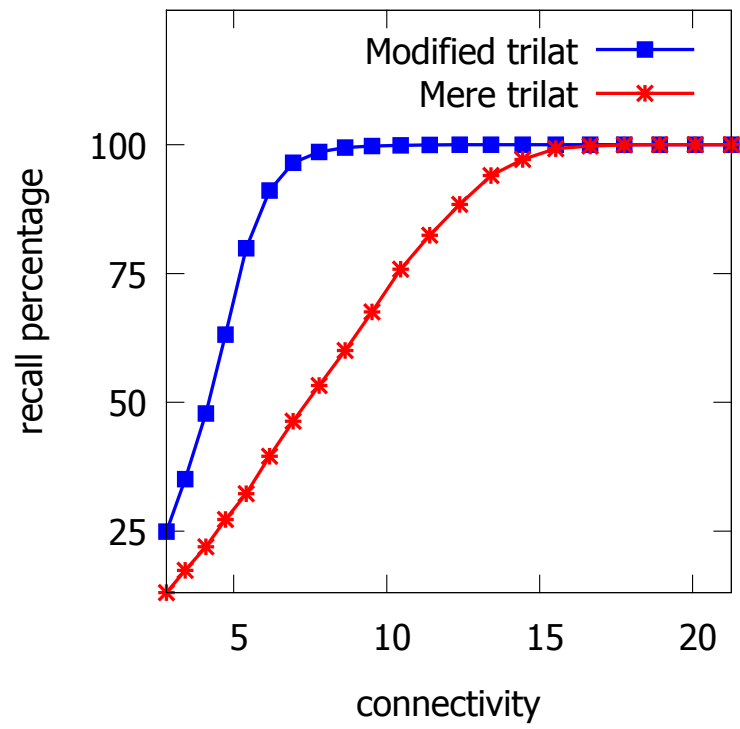


Figure 8: Recall percentage of trilateration when UDG violations are considered and when they are not.

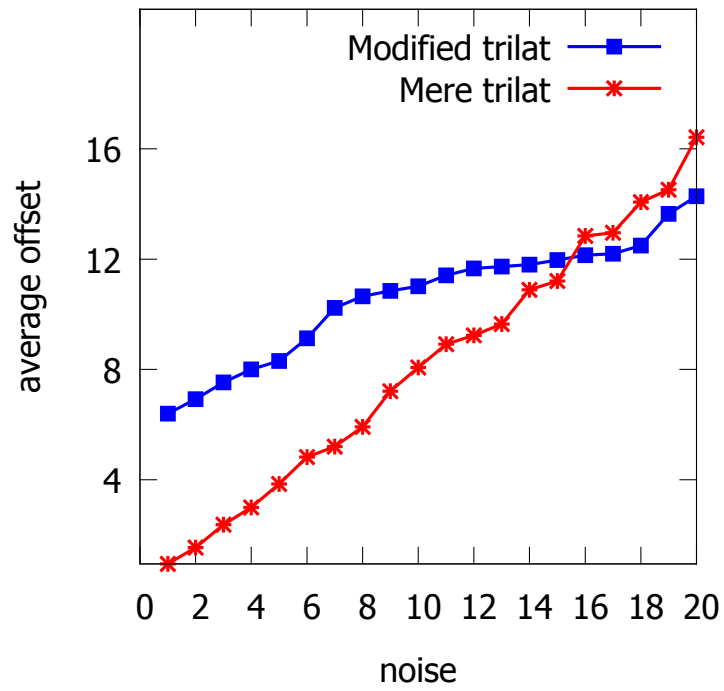


Figure 9: Average offset of trilateration when UDG violations are considered and when they are not.

References

- [1] S. Capkun, M. Hamdi, and J. Hubaux, “GPS-free positioning in mobile ad-hoc networks,” in *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, p. 10, 2001.
- [2] C. Chang, “Localization and Object-Tracking in an Ultrawideband Sensor Network,” Master’s thesis, UC Berkeley, USA, 2004.
- [3] H. Akcan and C. Evrendilek, “GPS-free directional localization via dual wireless radios,” *Computer Communications*, vol. 35, no. 9, pp. 1151–1163, 2012.
- [4] A. Pal, “Localization Algorithms in Wireless Sensor Networks: Current Approaches and Future Challenges,” *Network Protocols and Algorithms*, vol. 2, no. 1, pp. 45–73, 2010.
- [5] H. Akcan, V. Kriakov, H. Bronnimann, and A. Delis, “GPS-Free node localization in mobile wireless sensor networks,” in *Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE’06)*, (Chicago, Illinois, USA), p. 35–42, 2006.
- [6] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low-cost outdoor localization for very small devices,” *Personal Communications, IEEE*, vol. 7, pp. 28–34, Oct 2000.
- [7] G. Laman, “On graphs and rigidity of plane skeletal structures,” *Journal of Engineering Mathematics*, vol. 4, no. 10, pp. 331 – 340, 2002.
- [8] J. Saxe, “Embeddability of weighted graphs in k-space is strongly NP-Hard.,” in *17th Allerton Conference in Communications, Control and Computing on*, pp. 480–489, 1979.
- [9] J. Aspnes, D. Goldenberg, and R. Yang, “On the Computational Complexity of Sensor Network Localization,” in *In Proceedings of First International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, pp. 32–44, 2004.
- [10] T. Eren, O. Goldenberg, W. Whiteley, Y. R. Yang, A. Morse, B. D. O. Anderson, and P. Belhumeur, “Rigidity, computation, and randomization in network localization,” in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2673–2684, 2004.
- [11] J. Aspnes, T. Eren, D. Goldenberg, A. Morse, W. Whiteley, Y. Yang, B. Anderson, and P. Belhumeur, “A theory of network localization,” in *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1663–1678, 2010.
- [12] B. Alavi and K. Pahlavan, “Modeling of the TOA-based distance measurement error using UWB indoor radio measurements,” *Communications Letters, IEEE*, vol. 10, pp. 275–277, Apr 2006.
- [13] J. Park, E. Demaine, and S. Teller, “Moving-Baseline Localization,” in *Information Processing in Sensor Networks, 2008. IPSN ’08. International Conference on*, pp. 15–26, Apr 2008.